



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

popcorn global-computing

SEARCH

THE ACM DIGITAL LIBRARY




[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Term used popcorn

Found 53 of 143,484

Sort results
by

relevance

 Save results to a Binder

Try an Advanced Search

Try this search in [The ACM Guide](#)

Display results

expanded form



Search Tips

- ☐ Open results in a new window

Results 1 - 20 of 53


Result page: **1** 2 3 next

Relevance scale

¹ The POPCORN market—an online market for computational resources

Ori Regev, Noam Nisan

October 1998 **Proceedings of the first international conference on Information and computation economies**

Full text available:  pdf(1.12 MB)

Additional Information: [full citation](#), [references](#), [citing](#), [index terms](#)

Keywords: Internet, Java, global computation, markets, resource allocation

2 Sound: an emotional element of interactions a case study of a microwave oven

Cheong-Hyun Lee, Soony Kim, Choong-Seong Chae, Kook-Hyun Chung

August 2000 **Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques**

Full text available: pdf(539.14 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Little attention has been given to sound as an element of interaction although an interaction design utilizes five senses of the human being. While designing the portable microwave oven, we selected the sound as a topic of our study. As several sounds have already been applied to products, such as the buzz sound in the walk signal or a warning sound in the computer, the focus of research is that the sound may have to be different vis-...-vis the products functionality or loca ...

Keywords: design, element, emotions, interaction, microwave, outdoor environment, portable, sound, taste of food, users

3 Dynamic software updating

Michael Hicks, Jonathan T. Moore, Scott Nettles

May 2001 **ACM SIGPLAN Notices**, Proceedings of the ACM SIGPLAN 2001 conference
on Programming language design and implementation, Volume 36 Issue 5

Full text available: pdf(1.44 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Many important applications must run continuously and without interruption, yet must be changed to fix bugs or upgrade functionality. No prior general-purpose methodology for dynamic updating achieves a practical balance between flexibility, robustness, low

overhead, and ease of use.

We present a new approach for C-like languages that provides type-safe dynamic updating of native code in an extremely flexible manner (code, data, and types may be updated, at programmer-determined times ...

4 Plymouth Neon popcorn

Judy Conner

January 1996 **ACM SIGGRAPH 96 Visual Proceedings: The art and interdisciplinary programs of SIGGRAPH '96**

Additional Information: [full citation](#)



5 A cost-benefit framework for online management of a metacomputing system

Yair Amir, Baruch Awerbuch, R. Sean Borgstrom

October 1998 **Proceedings of the first international conference on Information and computation economies**

Full text available:  [pdf\(855.46 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)



Keywords: markets, metacomputing, networks, resource allocation

6 The special training needs of the first-time microcomputer user

Lisa Hines

December 1987 **Proceedings of the 15th annual ACM SIGUCCS conference on User Services**

Full text available:  [pdf\(906.69 KB\)](#) Additional Information: [full citation](#), [index terms](#)



7 A game-theoretic formulation of multi-agent resource allocation

Jonathan Bredin, Rajiv T. Maheswaran, Çağrı Imer, Tamer Başar, David Kotz, Daniela Rus

June 2000 **Proceedings of the fourth international conference on Autonomous agents**

Full text available:  [pdf\(735.87 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



8 Javelin++: scalability issues in global computing

Michael O. Neary, Sean P. Brydon, Paul Kmiec, Sami Rollins, Peter Cappello

June 1999 **Proceedings of the ACM 1999 conference on Java Grande**


Full text available:  [pdf\(1.34 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



9 Toward a foundational typed assembly language

Karl Crary

January 2003 **ACM SIGPLAN Notices , Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 38 Issue 1

Full text available:  [pdf\(299.12 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



We present the design of a typed assembly language called TALT that supports heterogeneous tuples, disjoint sums, and a general account of addressing modes. TALT also

implements the von Neumann model in which programs are stored in memory, and supports relative addressing. Type safety for execution and for garbage collection are shown by machine-checkable proofs. TALT is the first formalized typed assembly language to provide any of these features.

Keywords: proof-carrying code, typed assembly language

10 Safety checking of machine code

Zhichen Xu, Barton P. Miller, Thomas Reps

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5

Full text available:  [pdf\(306.71 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


We show how to determine statically whether it is safe for untrusted machine code to be loaded into a trusted host system. Our safety-checking technique operates directly on the untrusted machine-code program, requiring only that the initial inputs to the untrusted program be annotated with typestate information and linear constraints. This approach opens up the possibility of being able to certify code produced by any compiler from any source language, which gives the code prod ...

11 Fast forward: Metaphors and user interfaces in the 21st Century

Aaron Marcus

March 2002 **interactions**, Volume 9 Issue 2

Full text available:  [pdf\(121.22 KB\)](#)

 [html\(15.71 KB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

"Fast Forward" is a new column that the editors of *interactions* have kindly permitted me to write. It presents an opportunity for me to think about where our profession of user-interface design has been over the past three-and-a-half decades in which I have worked and about its future. I hope you will enjoy and benefit from joining me in this discussion. Your feedback is welcome.

12 Session 3: Mechanism design for online real-time scheduling

Ryan Porter

May 2004 **Proceedings of the 5th ACM conference on Electronic commerce**

Full text available:  [pdf\(256.48 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

For the problem of online real-time scheduling of jobs on a single processor, previous work presents matching upper and lower bounds on the competitive ratio that can be achieved by a deterministic algorithm. However, these results only apply to the non-strategic setting in which the jobs are released directly to the algorithm. Motivated by emerging areas such as grid computing, we instead consider this problem in an economic setting, in which each job is released to a separate, self-interested ...

Keywords: game theory, mechanism design, online algorithms, scheduling

13 A dependently typed assembly language

Hongwei Xi, Robert Harper

October 2001 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN international conference on Functional programming**, Volume 36 Issue 10

Full text available:  [pdf\(271.30 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


We present a dependently typed assembly language (DTAL) in which the type system

supports the use of a restricted form of dependent types, reaping some benefits of dependent types at the assembly level. DTAL improves upon TAL, enabling certain important compiler optimizations such as run-time array bound check elimination and tag check elimination. Also, DTAL formally addresses the issue of representing sum types at assembly level, making it suitable for handling not only datatypes in ML but al ...

14 Two-Handed Interaction: Comparing voodoo dolls and HOMER: exploring the importance of feedback in virtual environments

Jeffrey S. Pierce, Randy Pausch

April 2002 **Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves**

Full text available:  pdf(386.20 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


When creating techniques for manipulating objects at a distance in immersive virtual environments, researchers have primarily focused on increasing selection range, placement range, and placement accuracy. This focus has led researchers to create and formally study a series of "arm-extension" techniques, which dynamically scale the user's arm to allow him to manipulate distant objects. Researchers have also developed representation-based techniques, which allow users to manipulate a distant obje ...

Keywords: 3D interaction, object manipulation, virtual reality

15 A certifying compiler for Java

Christopher Colby, Peter Lee, George C. Necula, Fred Blau, Mark Plesko, Kenneth Cline

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5


Full text available:  pdf(792.48 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citing terms](#), [index terms](#)

This paper presents the initial results of a project to determine if the techniques of proof-carrying code and certifying compilers can be applied to programming languages of realistic size and complexity. The experiment shows that: (1) it is possible to implement a certifying native-code compiler for a large subset of the Java programming language; (2) the compiler is freely able to apply many standard local and global optimizations; and (3) the PCC bina ...

16 Translation validation for an optimizing compiler

George C. Necula

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5

Full text available:  pdf(679.20 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citing terms](#), [index terms](#)

We describe a translation validation infrastructure for the GNU C compiler. During the compilation the infrastructure compares the intermediate form of the program before and after each compiler pass and verifies the preservation of semantics. We discuss a general framework that the optimizer can use to communicate to the validator what transformations were performed. Our implementation however does not rely on help from the optimizer and it is quite successful by using instead a few heuri ...

17 Resource bound certification

Karl Crary, Stephnie Weirich

January 2000 **Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Full text available:  pdf(1.71 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citing terms](#), [index terms](#)

Various code certification systems allow the certification and static verification of important safety properties such as memory and control-flow safety. These systems are valuable tools for verifying that untrusted and potentially malicious code is safe before execution. However, one important safety property that is not usually included is that programs adhere to specific bounds on resource consumption, such as running time. We present a decidable type system capable of specify ...

18 Type dispatch for named hierarchical types

Neal Glew

September 1999 **ACM SIGPLAN Notices , Proceedings of the fourth ACM SIGPLAN international conference on Functional programming**, Volume 34 Issue 9

Full text available:  pdf(1.34 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Type dispatch constructs are an important feature of many programming languages. Scheme has predicates for testing the runtime type of a value. Java has a class cast expression and a try statement for switching on an exception's class. Crucial to these mechanisms, in typed languages, is type refinement: The static type system will use type dispatch to refine types in successful branches. Considerable previous work has addressed type case constructs for structural type systems without subtyping, ...

19 A componentized architecture for dynamic electronic markets

Benny Reich, Israel Ben-Shaul

December 1998 **ACM SIGMOD Record**, Volume 27 Issue 4

Full text available:  pdf(1.13 MB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The emergence and growing popularity of Internet-based electronic market-places, in their various forms, has raised the challenge to explore genericity in market design. In this paper we present a domain-specific software architecture that delineates the abstract components of a generic market and specifies control and data-flow constraints between them, and a framework that allows convenient pluggability of components that implement specific market policies. The framework was realized in t ...

20 Hypertextual dynamics in a life set for two

Robert Kendall

March 1996 **Proceedings of the the seventh ACM conference on Hypertext**

Full text available:  pdf(1.22 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: dynamic links, embedded variables, floating links, global states, poetry, reading templates, variable nodes

Results 1 - 20 of 53

Result page: [1](#) [2](#) [3](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

L Number	Hits	Search Text	DB	Time stamp
1	8341	707/\$.ccls 709/225 709/200/ 709/217	USPAT	2004/10/21 15:48
2	2	(707/\$.ccls 709/225 709/200/ 709/217) and (distributed with query with process\$3) and optimi\$6 and performance and internet and bandwidth	USPAT	2004/10/21 15:49
-	234	distributed with query with process\$3	USPAT	2004/10/01 19:34
-	32	(distributed with query with process\$3) and optimi\$6 and performance and internet	USPAT	2004/10/21 15:42
-	1	((distributed with query with process\$3) and optimi\$6 and performance and internet) and (multi\$thread\$3 multi adj thread\$3)	USPAT	2004/09/08 18:49
-	110	distributed near4 query near4 process\$3	USPAT	2004/09/08 18:51
-	18	distributed near4 query near4 process\$3 and (parallel adj processing)	USPAT	2004/09/08 18:53
-	5	(distributed near4 query near4 process\$3) same (parallel adj processing)	USPAT	2004/09/08 18:55
-	3	(distributed near4 query near4 process\$3) same (parallel adj processing) and work and load\$1	USPAT	2004/09/08 18:56
-	6	(distributed near4 query near4 process\$3) same (parallel adj processing) and work and load\$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/08 18:56
-	18	(distributed near4 query near4 process\$3) and (parallel adj processing) and work and load\$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/08 18:58
-	14	(distributed near4 query near4 process\$3) and (parallel adj processing) and work and load\$1 and sql	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/08 18:58
-	4	(distributed near4 query near4 process\$3) and (parallel adj processing) and work and load\$1 and sql and thread\$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/08 18:58
-	13	("4669730" "4815741" "5056019" "5332218" "5483444" "5598566" "5703949" "5802062" "5815793" "5826265" "5862325" "5884072" "5909540" "5916024" "5918229" "5918229" "5921865" "5937192" "598010" "5966451" "1970477" "5987056" "6003065" "6009455" "6014634" "6024640" "6026474" "6052785" "6058393" "6061660" "6070190" "6094654" "6112225" "6112181" "6115713" "6148377" "6148377" "6151684" "5964382" "5655081" "5978594" "5740549") and (distributed near3 query)	USPAT	2004/10/01 19:42
-	0	("4669730" "4815741" "5056019" "5332218" "5483444" "5598566" "5703949" "5802062" "5815793" "5826265" "5862325" "5884072" "5909540" "5916024" "5918229" "5918229" "5921865" "5937192" "598010" "5966451" "1970477" "5987056" "6003065" "6009455" "6014634" "6024640" "6026474" "6052785" "6058393" "6061660" "6070190" "6094654" "6112225" "6112181" "6115713" "6148377" "6148377" "6151684" "5964382" "5655081" "5978594" "5740549") and (distributed near3 query) same (thread\$4 worker)	USPAT	2004/10/01 19:43

-	4	("4669730" "4815741" "5056019" "5332218" "5483444" "5598566" "5703949" "5802062" "5815793" "5826265" "5862325" "5884072" "5909540" "5916024" "5918229" "5918229" "5921865" "5937192" "598010" "5966451" "1970477" "5987056" "6003065" "6009455" "6014634" "6024640" "6026474" "6052785" "6058393" "6061660" "6070190" "6094654" "6112225" "6112181" "6115713" "6148377" "6148377" "6151684" "5964382" "5655081" "5978594" "5740549") and (distributed near3 query) and (thread\$4 worker)	USPAT	2004/10/19 18:12
-	2	("5958009" "6633898") .PN.	USPAT	2004/10/19 18:13
-	5	("5768532" "5907619" "5953420" "6135646" "64991532")	USPAT .PN.	2004/10/19 18:13
-	1532	"4669730" "4815741" "5056019" "5332218" "5483444" "5598566" "5703949" "5802062" "5815793" "5826265" "5862325" "5884072" "5909540" "5916024" "5918229" "5918229" "5921865" "5937192" "598010" "5966451" "1970477" "5987056" "6003065" "6009455" "6014634" "6024640" "6026474" "6052785" "6058393" "6061660" "6070190" "6094654" "6112225" "6112181" "6115713" "6148377" "6148377" "6151684" "5964382" "5655081" "5978594" "5740549")	USPAT	2004/10/19 18:20
-	3	("4669730" "4815741" "5056019" "5332218" "5483444" "5598566" "5703949" "5802062" "5815793" "5826265" "5862325" "5884072" "5909540" "5916024" "5918229" "5918229" "5921865" "5937192" "598010" "5966451" "1970477" "5987056" "6003065" "6009455" "6014634" "6024640" "6026474" "6052785" "6058393" "6061660" "6070190" "6094654" "6112225" "6112181" "6115713" "6148377" "6148377" "6151684" "5964382" "5655081" "5978594" "5740549") and (distributed with query with process\$3)	USPAT	2004/10/20 18:57
-	338	("4669730" "4815741" "5056019" "5332218" "5483444" "5598566" "5703949" "5802062" "5815793" "5826265" "5862325" "5884072" "5909540" "5916024" "5918229" "5918229" "5921865" "5937192" "598010" "5966451" "1970477" "5987056" "6003065" "6009455" "6014634" "6024640" "6026474" "6052785" "6058393" "6061660" "6070190" "6094654" "6112225" "6112181" "6115713" "6148377" "6148377" "6151684" "5964382" "5655081" "5978594" "5740549") and bandwidth	USPAT	2004/10/20 18:58
-	11	("4669730" "4815741" "5056019" "5332218" "5483444" "5598566" "5703949" "5802062" "5815793" "5826265" "5862325" "5884072" "5909540" "5916024" "5918229" "5918229" "5921865" "5937192" "598010" "5966451" "1970477" "5987056" "6003065" "6009455" "6014634" "6024640" "6026474" "6052785" "6058393" "6061660" "6070190" "6094654" "6112225" "6112181" "6115713" "6148377" "6148377" "6151684" "5964382" "5655081" "5978594" "5740549") and bandwidth and optimize and performance and (disk near3 usage)	USPAT	2004/10/20 18:59
-	13	(distributed with query with process\$3) and optimi\$6 and performance and internet and bandwidth	USPAT	2004/10/20 19:00